

# EDGE DETECTION MENGGUNAKAN METODE *ROBERTS' CROSS*

Arifin<sup>1</sup>, Budiman<sup>2</sup>

STMIK Mikroskil

Jl. Thamrin No. 112, 124, 140 Medan 20212

arifinS2C@yahoo.com<sup>1</sup>, sync\_vlo@yahoo.com<sup>2</sup>

---

## Abstrak

Pengolahan citra digital merupakan satu cabang ilmu pengetahuan yang saat ini banyak dikembangkan demi mempermudah kehidupan manusia. Salah satunya adalah pendeteksian tepi (*Edge Detection*), yang banyak digunakan untuk mendeteksi/mengenal data *biometric* seperti sidik jari, wajah, retina, sampai ke bidang kedokteran seperti untuk diagnosa kanker. Ada banyak metode *Edge Detection* yang telah diajukan oleh para pakar ilmu tersebut, seperti *Sobel*, *Scharr*, *Prewitt*, *Roberts' Cross*, dan sebagainya. Masing-masing memiliki kelebihan maupun kekurangan sendiri. Metode *Roberts' Cross* memiliki tingkat kekaburan orthogonal yang kecil serta resolusi yang baik sehingga banyak digunakan dibandingkan metode lainnya. Penulis mengupas dan mengimplementasikan *Edge Detection* dengan metode *Roberts' Cross* dan memperlihatkan hasil eksekusi metode tersebut pada beberapa citra.

**Kata kunci :** *Edge Detection, Pengolahan Citra, Roberts Cross*

---

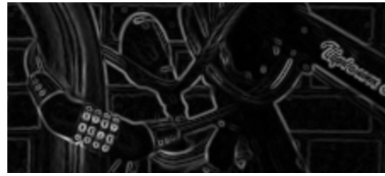
## 1. Pendahuluan

*Edge Detection* merupakan salah satu proses yang fundamental dalam Pengolahan Citra yang bertujuan mengidentifikasi titik-titik pada citra digital dimana tingkat kecerahan (*brightness*) berubah drastis atau terjadi diskontinuitas. *Edge Detection* bertujuan membentuk sejumlah kurva yang saling terhubung yang mengindikasikan batas-batas objek, tanda-tanda permukaan, serta kurva-kurva yang mengindikasikan diskontinuitas pada orientasi permukaan. Penerapan *Edge Detection* secara signifikan dapat mengurangi jumlah data yang diproses dan menyaring informasi yang mungkin dianggap kurang relevan, sambil menjaga sifat struktural penting dari suatu gambar. *Edge Detection* banyak digunakan dalam aplikasi-aplikasi pengeditan image.



Gambar 1.1 Citra Asli [9]

Gambar 1.1 dan 1.2 menunjukkan perbandingan dari hasil penerapan metode-metode *Edge Detection* terhadap sebuah citra input yang sama (perhatikan tulisan “*Uptown*” yang ada di tiang di bagian kanan citra). Di antara metode-metode tersebut, metode *Roberts' Cross* memiliki tingkat kekaburan orthogonal yang lebih kecil serta resolusi yang lebih baik, dan penulis memilih mengupas dan mengimplementasikan metode tersebut.



Metode *Sobel*



Metode *Scharr*



Metode *Prewitt*



Metode *Roberts' Cross*

Gambar 1.2 Perbandingan Hasil *Edge Detection* pada Beberapa Metode [9]

Tujuan dari penulisan bedah paper oleh penulis adalah:

1. Mengupas metode *Roberts' Cross* sebagai salah satu metode dalam *Edge Detection System* dan menerapkan metode tersebut menjadi algoritma yang siap digunakan.
2. Menunjukkan hasil penerapan metode *Roberts' Cross* pada beberapa citra.

Sementara manfaat yang bisa didapatkan dari penelitian ini antara lain:

1. Meningkatkan pengetahuan pembaca, terutama pelaku IT tentang manfaat *Edge Detection* terutama dalam pengenalan wajah, pengenalan retina mata dan karakteristik biometric lainnya, pendeteksi kanker melalui analisa hasil *rontgen*, dan bidang- bidang lainnya yang jumlahnya terus bertambah.
2. Algoritma yang dihasilkan dapat digunakan dan implementasikan bagi pembaca yang ingin menerapkan metode *Roberts' Cross*.

Penulis melakukan analisa cara kerja metode *Roberts' Cross* dan merancang algoritmanya menggunakan citra 2 dimensi dengan warna 32 bit dan berformat jpg dan bmp. Penulis juga melakukan pengujian pada empat buah citra yang masing-masing memiliki karakteristik tersendiri.

## 2. Kajian Pustaka

### 2.1 Citra Digital

Citra digital merupakan sebuah fungsi intensitas cahaya  $f(x,y)$ , dimana nilai  $x$  dan  $y$  merupakan koordinat spasial dan nilai fungsi tersebut pada setiap titik  $(x,y)$  merupakan tingkat kecermerlangan citra pada titik tersebut. Citra digital dinyatakan dengan matrik berukuran  $N \times M$  (  $N$  menyatakan baris atau tinggi,  $M$  menyatakan kolom atau lebar ) [4]. Ukurannya dinyatakan dalam titik atau piksel (*pixel* = picture element) dan dapat pula dinyatakan dalam satuan panjang (mm atau inci = *inch*).

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Keterangan:

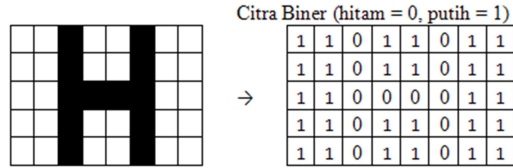
$N$  = jumlah baris,  $0 \leq y \leq N - 1$

$M$  = jumlah kolom,  $0 \leq x \leq M - 1$

$L$  = maksimal warna intensitas (derajat keabuan),  $0 \leq f(x,y) \leq L - 1$

Berdasarkan format penyimpanan nilai warnanya, citra terdiri atas empat jenis [4]:

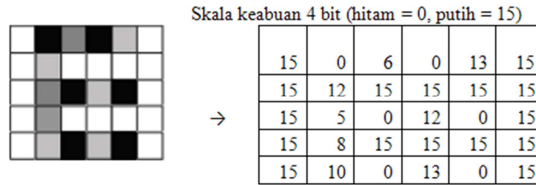
1. Citra biner atau monokrom



Gambar 2.1 Citra Asli

Pada citra jenis ini, setiap titik atau piksel hanya bernilai 0 atau 1. Gambar 2.1 merupakan contoh citra biner.

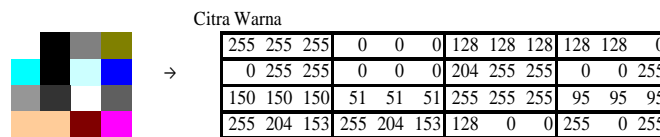
2. Citra skala keabuan



Gambar 2.2 Citra Skala Keabuan

Citra skala keabuan mempunyai kemungkinan warna antara hitam (minimal) dan putih (maksimal). Suatu citra dengan skala keabuan 4 bit, memiliki jumlah kemungkinan warna  $2^4$  16 warna. Gambar 2.2 memperlihatkan citra skala keabuan 4 bit.

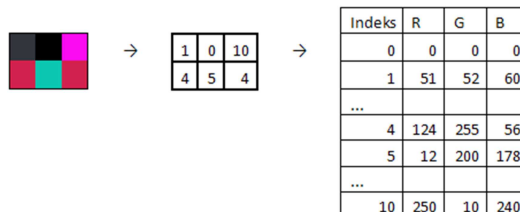
3. Citra warna (*true color*)



Gambar 2.3 Citra Warna

Setiap titik (piksel) pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar yaitu merah, hijau, dan biru yang dikenal sebagai citra RGB (*Red, Green, Blue*). Setiap warna dasar memiliki intensitas sendiri dengan nilai maksimum 255 (8 bit). Gambar 2.3 memperlihatkan contoh citra warna.

4. Citra warna berindeks



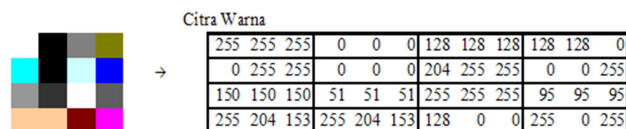
Gambar 2.4 Citra Warna Berindeks

Setiap titik (piksel) pada citra warna berindeks mewakili indeks dari suatu tabel warna yang tersedia (biasanya disebut palet warna). Contoh citra warna berindeks diperlihatkan oleh Gambar 2.4.

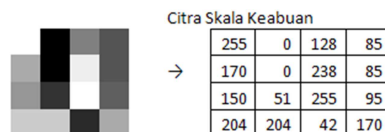
## 2.2 Konversi Citra Warna (RGB) ke Citra Skala Keabuan (*Grayscale*)

Proses pertama dari metode pendeteksi citra secara umum termasuk metode *Roberts' Cross* adalah melakukan konversi citra, dari citra berwarna ke citra skala keabuan (*grayscale*). Rumus yang digunakan untuk konversi adalah sebagai berikut:  $x = \frac{(R+G+B)}{3}$ , dimana R adalah nilai warna merah (*Red*), G adalah nilai warna hijau (*Green*), dan B adalah nilai warna biru (*Blue*) dari *pixel* yang diproses.

Gambar 2.6 menunjukkan hasil proses konversi citra awal pada gambar 2.5 yang merupakan sebuah citra warna.



Gambar 2.5 Citra Warna



Gambar 2.6 Citra Hasil Konversi Citra Warna ke Citra Skala Keabuan

## 2.3 Resolusi Citra

Resolusi citra adalah banyaknya titik untuk setiap satuan panjang (*dot per inch*) yang terdapat dalam citra tersebut. Semakin besar resolusi makin banyak titik yang terkandung dalam citra, sehingga menjadi lebih halus dalam visualisasinya dan semakin berkualitas[4].

Menurut kualitas resolusi citra yang dihasilkan, resolusi dibedakan [7]:

1. Resolusi tinggi (*high resolution*), jika tingkat ketelitian yang cukup tinggi dari suatu media dalam menangkap ataupun menampilkan datanya.
2. Resolusi rendah (*low resolution*), jika pada gambar atau layar yang teks dan grafiknya tampil dengan detail yang relatif kasar.

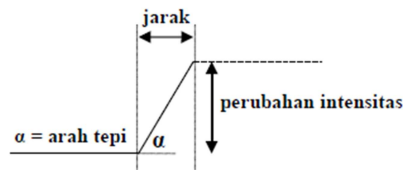
Gambar 2.7 memperlihatkan gambar suatu objek yang dicitrakan menggunakan beberapa kamera dengan tingkat kualitas resolusi yang berbeda.



Gambar 2.7 Citra dengan Beberapa Tingkat Resolusi

## 2.5 Edge Detection

*Edge* dapat didefinisikan sebagai batas antara dua *region* (dua piksel yang saling berdekatan) yang memiliki perbedaan intensitas yang tajam atau tinggi (Febriani, 2008). *Edge* dapat diorientasikan dengan suatu arah, dan arah ini berbeda-beda, tergantung pada perubahan intensitas. Gambar 2.8 memperlihatkan model *edge* dalam ruang satu dimensi.



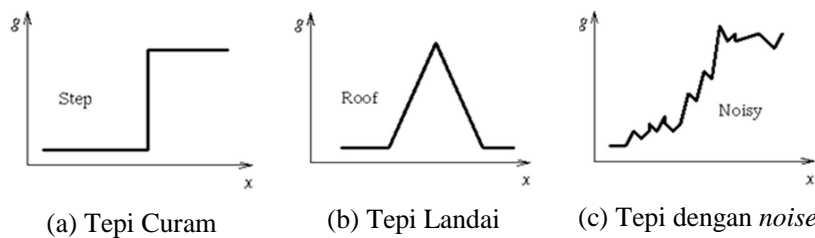
Gambar 2.8 Model *Edge* Satu Dimensi

Menurut Munir [7] ada tiga macam tepi (*edge*) yang terdapat di dalam citra digital seperti yang diperlihatkan pada gambar 2.9. Ketiga macam *edge* tersebut adalah:

1. Tepi curam, di mana tepi terbentuk karena perubahan intensitas yang tajam, berkisar  $90^\circ$ .
2. Tepi landai, di mana tepi lebar, sudut arah kecil. Terdiri dari sejumlah tepi-tepi lokal yang lokasinya berdekatan.
3. Tepi yang mengandung *noise*, di mana biasanya dilakukan operator *image enhancement* terlebih dahulu, misalnya Operator Gaussian yang berfungsi untuk menghaluskan citra.

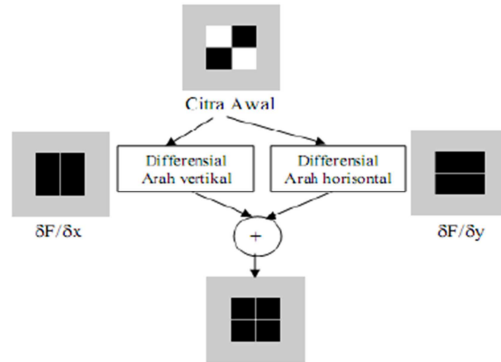
*Edge detection* pada suatu citra memiliki tujuan sebagai berikut [3]:

- a. Menandai bagian yang menjadi detil citra.
- b. Memperbaiki detil citra yang kabur karena *error* atau efek proses akuisisi.



Gambar 2.9 Jenis-jenis Tepi

Gambar 2.10 memperlihatkan bagaimana *edge* dari suatu citra dapat diperoleh dengan operasi *edge detection*. Berdasarkan prinsip-prinsip filter pada citra, *edge* suatu gambar dapat diperoleh menggunakan *High Pass Filter* (HPF), dengan karakteristik:  $\sum \sum H(x, y) = 0$  (2.1)



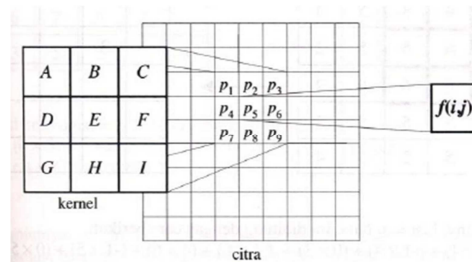
Gambar 2.10 Proses *Edge Detection* Citra

## 2.6 Konvolusi

*Edge detection* merupakan salah satu proses pengolahan citra yang menggunakan filter, yang dilakukan melalui metode konvolusi. Konvolusi dinyatakan dalam matriks, dimana setiap elemen matriks dinamakan koefisien konvolusi. Operasi konvolusi bekerja dengan menggeser kernel piksel per piksel, yang hasilnya disimpan dalam matriks baru. Konvolusi 2 fungsi  $f(x)$  dan  $g(x)$  diperlihatkan dengan rumus berikut:  $f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha)g(x-\alpha)d\alpha$  atau fungsi diskritnya:  $f(x) * g(x) = \sum f(\alpha)g(x-\alpha)$ , dimana  $\alpha$  = peubah bantu.

Pada konvolusi 2D fungsi malarnya dapat dihitung dengan persamaan:  $f(x, y) * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b)g(x-a, y-b)dadb$ . Sedangkan fungsi diskritnya dihitung dengan persamaan:  $f(x, y) * g(x, y) = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} f(a, b)g(x-a, y-b)$ .  $G(x)$  merupakan *convolution mask/filter/kernel* atau *template*. Gambar 2.11 memperlihatkan ilustrasi terjadinya konvolusi, di mana:  $f(i, j) = Ap_1 + Bp_2 + Cp_3 + Dp_4 + Ep_5 + Fp_6 + Gp_7 + Hp_8 + Ip_9$ .

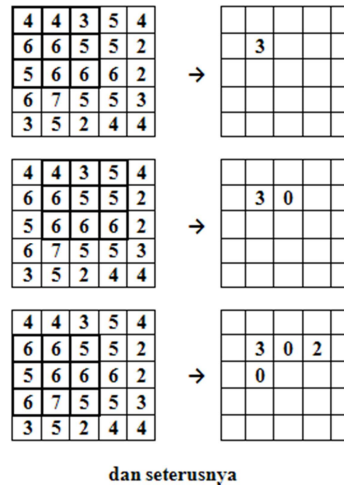
Untuk lebih jelasnya, berikut contoh konvolusi yang terjadi antara citra  $f(x, y)$  berukuran 5x5 dengan sebuah kernel berukuran 3x3 yang diperlihatkan oleh Gambar 2.12. Tanda ■ menunjukkan posisi (0,0) dari kernel.



Gambar 2.11 Proses Konvolusi

$$f(x) = \begin{bmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{bmatrix} \quad g(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Gambar 2.12 Matriks Citra dan Kernel Sebelum Konvolusi



Gambar 2.13 Tahapan Proses Pembentukan Konvolusi

		3	0	2
	0	2	6	
	6	0	2	

Gambar 2.14 Hasil Konvolusi Citra dan Kernel

Proses perkalian matriks pada setiap titik pixel yang bersesuaian pada matriks citra dengan matriks kernel. Tahapan untuk mendapatkan hasil konvolusi yang terjadi antara citra dan kernel di atas dapat dilihat pada Gambar 2.13, sehingga diperoleh hasil akhir dari proses konvolusi tersebut, yang ditunjukkan oleh Gambar 2.14.

Seperti yang bisa diperhatikan pada Gambar 2.14, hasil akhir dari proses konvolusi mengalami pengurangan dua pixel secara horizontal maupun vertikal. Hal ini terjadi karena matriks kernel tidak bisa mengakomodasikan perhitungan pada pixel-pixel paling tepi dari citra awal. Walaupun demikian, pixel-pixel paling tepi dianggap tidak mengandung informasi yang cukup penting sehingga penghilangan pixel-pixel tersebut dari citra hasil konvolusi tidak akan memberikan pengaruh yang signifikan.

Dalam konvolusi terdapat dua kemungkinan yang jika ditemukan, diselesaikan dengan cara berikut, yaitu:

1. Untuk hasil konvolusi dengan nilai negatif, nilainya dijadikan nol (0).
2. Jika hasil konvolusi lebih besar dari derajat keabuan maksimum, maka nilai diubah menjadi nilai maksimum derajat keabuan.

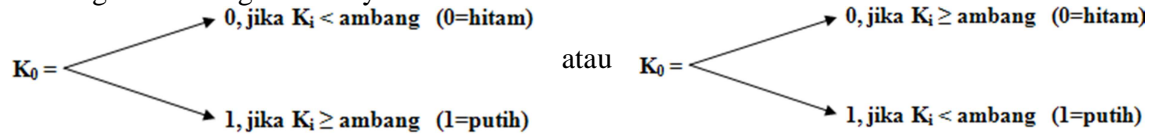
## 2.7 Thresholding

Operasi *thresholding* merupakan salah satu operasi yang termasuk ke dalam operasi titik dalam pengolahan citra digital. Operasi ini digunakan untuk mengubah citra dengan format skala keabuan yang memiliki kemungkinan nilai lebih dari 2 ke citra biner yang memiliki 2 buah nilai (yaitu 0 dan 1).

*Thresholding* terdiri dari dua jenis, yaitu:

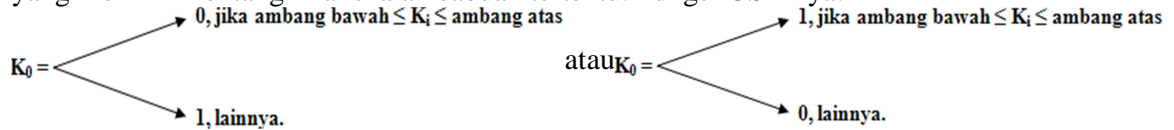
### 1. *Thresholding* Tunggal

*Thresholding* tunggal merupakan proses *thresholding* yang hanya memiliki sebuah nilai ambang batas. Fungsi GST-nya:



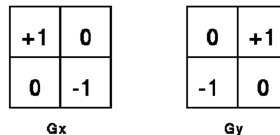
### 2. *Thresholding* Ganda

Memiliki ambang batas atas dan batas bawah. Dilakukan untuk menampilkan titik-titik yang memiliki rentang nilai skala keabuan tertentu. Fungsi GST-nya:



## 2.5.3 Metode *Edge Detection Roberts' Cross*

Pada teorinya, operator *Roberts' Cross* terdiri dari sepasang matriks konvolusi  $2 \times 2$ . Satu matriks merupakan hasil rotasi 90 derajat terhadap matriks lainnya. Operator sederhana ini memeriksa sebuah piksel tambahan pada satu arah gradient tetapi karena yang diperiksa adalah piksel dalam arah diagonal, maka secara keseluruhan piksel-piksel yang terlibat membentuk jendela matrik  $2 \times 2$ . Bentuk jendela yang demikian lebih menekankan pemeriksaan pada kedua arah diagonal, dari pada arah horizontal atau arah vertikal, sehingga perbedaan yang terletak pada sisi-sisi miring objek akan terdeteksi dengan lebih baik.



Gambar 2.15 Matriks *Kernel Roberts' Cross*

Matriks ini didesain agar dapat merespon dengan maksimal pada tepi yang bersudut  $45^\circ$  ke batas piksel, satu matriks untuk dua orientasi sudut  $90^\circ$ . Matriks ini dapat digunakan secara terpisah pada gambar masukan, untuk menghasilkan pengukuran yang terpisah dari komponen gradien pada setiap orientasi (Gx dan Gy). Matriks-matriks ini dapat selanjutnya dikombinasikan untuk mencari skala absolut dari gradien pada setiap titik dan orientasi dari gradien tersebut. Nilai mutlak gradien pada suatu titik dapat dicari dengan persamaan sebagai berikut:  $|G| = \sqrt{Gx^2 + Gy^2}$ . Sedangkan untuk perhitungan pendekatan gradien dapat dilakukan dengan menggunakan:  $|G| = |Gx| + |Gy|$  yang dapat diproses dengan lebih cepat.



### 3. Metode Penelitian

Metodologi yang digunakan penulis adalah metodologi *waterfall* dengan tahapan-tahapan sebagai berikut:

1. Pengumpulan data melalui buku, artikel, jurnal, makalah, dan situs internet sebagai referensi penelitian
2. Analisis dan Perancangan algoritma
  - Mempelajari dan menganalisis cara kerja metode *Roberts' Cross*.
  - Merancang algoritma berdasarkan hasil analisa
3. Coding, dengan menerapkan algoritma ke dalam sebuah program sederhana
4. Testing, dengan melakukan ujicoba terhadap program tersebut untuk membuktikan kebenaran algoritma

### 4. Hasil dan Pembahasan

<pre>For i=0 to Image.width   For j=0 to Image.height     Pixel=Image.GetPixel(i,j)     Warna=(P.R+P.G+P.B)/3     Image.SetPixels(i,j, Warna)   End For End For</pre>	<pre>For i = 0 to Image.width   For j = 0 to Image.height     Gx=Abs(Image.GetPixel(i,j)-     Image.GetPixels(i+1,j+1))     Gy=Abs(Image.GetPixel(i,j+1)     ) - Image.GetPixels(i+1,j))     GMutlak=Sqrt(Gx^2 + Gy^2)   End For End For</pre>	<pre>For i = 0 to Image.width   For j = 0 to Image.height     If GMutlak &lt; 0 then       GrMutlak = 0     If GMutlak &gt; 255 then       GradienMutlak = 255       SetPixel(i,j,RGB(GMutlak,       GMutlak, GMutlak))     End For   End For End For</pre>
(a)	(b)	(c)

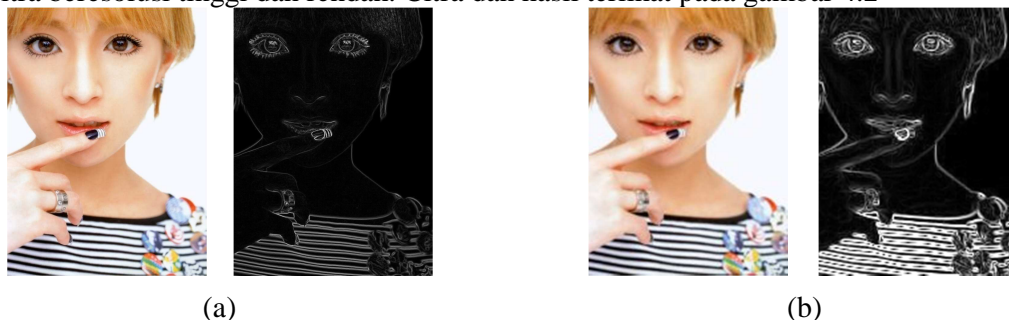
Gambar 4.1 (a) Algoritma Konversi ke Grayscale (b) Algoritma Konvolusi (c) Algoritma *Thresholding*

Metode *Roberts' Cross* dijalankan dengan tiga tahapan, dengan algoritma masing-masing tahapan terlihat pada gambar 4.1. Ketiga tahapan tersebut antara lain:

1. Konversi citra warna ke citra skala keabuan (*grayscale*).
2. Proses konvolusi citra.
3. Proses *thresholding*.

Penulis melakukan pengujian pada citra yang memiliki karakteristik tersendiri, yaitu:

1. Citra beresolusi tinggi dan rendah. Citra dan hasil terlihat pada gambar 4.2



Gambar 4.2 Perbandingan Hasil *Edge Detection* pada Citra dengan Resolusi dan Rendah

(a) Citra dengan Resolusi 600x807 Pixel. (b) Citra dengan Resolusi 150x202 Pixel.

2. Citra dengan tingkat kontras tinggi dan rendah. Citra dan hasil terlihat pada gambar 4.3



Gambar 4.3 Perbandingan Hasil *Edge Detection* pada Citra Berlainan Tingkat Kekontrasan  
(a) Citra dengan Kekontrasan Tinggi. (b) Citra dengan Kekontrasan Rendah

## 5. Kesimpulan

Kesimpulan yang bisa diambil dari kesimpulan ini adalah:

1. Semakin tinggi resolusi citra, semakin tajam dan halus *edge* yang dihasilkan pada citra output. Pada citra resolusi rendah, *edge* yang dihasilkan kurang tajam dan kelihatan kasar.
2. Semakin tinggi kontras citra, semakin baik pula kualitas dan jumlah *edge* yang dihasilkan pada citra output.
3. Metode *Roberts' Cross* menggunakan dua buah matriks konvolusi berukuran kecil yaitu 2x2. Hal ini menyebabkan jumlah proses perkalian menjadi lebih sedikit, tetapi rawan terhadap *noise* atau *pixel* rusak.

## Referensi

- [1] Febriani. 2008. "Analisis penelusuran tepi citra menggunakan deteksi tepi sobel dan canny". Dalam Febriani dan Lussiana, E. T. P. (eds). *Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008)*: hal 462-466. Depok: Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Gunadarma.
- [2] Indira, M. 2008. "Perbandingan metode pendeteksi tepi". Dalam Indira, M, dkk. (eds). *Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008)*: hal 366-373. Depok: Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Gunadarma.
- [3] Sigit, R, *et al.* 2005. *Step By Step Pengolahan Citra Digital*. Yogyakarta : Andi Offset.
- [4] Hestningsih, I. 2008. *Pengolahan Citra*.
- [5] Taurisna, A. 2009. "Analisis pengaruh kualitas resolusi citra terhadap kinerja metode pendeteksi tepi". Medan : Departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sumatera Utara.
- [6] Shafry, M., Rahim, M., Isrozaidi, N., Ismail, N., dan Idris, M. A. S. 2004. "The use of hybrid technique: thresholding and edge detection for identifying river from aerial photo". *Jurnal Teknologi*, 41(B) Dis. 2004: 1–20. Universiti Teknologi Malaysia
- [7] <http://www.total.or.id/info.php/kk=Resolusi>.
- [8] [http://en.wikipedia.org/wiki/Edge\\_detection](http://en.wikipedia.org/wiki/Edge_detection).
- [9] [http://en.wikipedia.org/wiki/Roberts\\_Cross](http://en.wikipedia.org/wiki/Roberts_Cross).